

**REMARKS**

Claims 1-24 are pending in the present application. Claims 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21 and 24 have been amended herewith. Reconsideration of the claims is respectfully requested.

Applicants would initially like to thank the Examiner for taking the time to conduct a telephone interview for this case on 07/20/2004. While no agreement was reached, Applicants representative emphasized that the invention was specifically directed to an ability to debug *partition management* firmware.

**I. 35 U.S.C. § 103, Obviousness**

The Examiner rejected Claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable over Held, US 5,889,988 in view of Bouchier, US 6,684,343. This rejection is respectfully traversed.

With respect to Claim 1, none of the cited references teach or suggest the claimed steps of "establishing extensions within said operating system debugger, said extensions enabling direct access to the resources of the partition management firmware by said operating system debugger" and "utilizing said extensions to debug said partition management firmware". In rejecting Claim 1, the Examiner states that the cited Held reference teaches establishing extensions within operating system debugger, and that the cited Bouchier discloses partition management firmware that is debugged. Applicants show that Claim 1 goes beyond mere recitation of debugger extensions and debugging partition management firmware. Specifically, there is a synergistic linkage between these two claimed features, whereby the debugger extensions are used to debug the partition management firmware. This claimed feature advantageously allows for use of existing operating system debuggers to debug partition management firmware (Specification page 3, lines 13-20; page 4, lines 12-20, et seq.). The extensions as taught by Held do not provide any ability to debug partition management firmware, as there is no debugging of partition management firmware by any means in the teachings of Held. The debugger as taught by Bouchier is a dedicated, standalone debugger with no extensions (Col. 12, lines 59-60; FIG. 8). Thus, it is urged that none of the cited references teach or suggest establishing extensions within the operating system debugger and utilizing these

extensions to debug the partition management firmware. Thus, Claim 1 is shown to not be obvious in view of the cited references, as all claimed limitations are not taught or suggested by the cited references<sup>1</sup>.

Further with respect to Claim 1, none of the cited references teach or suggest the claimed feature of "debug said partition management firmware". The Examiner cites Bouchier as teaching this claimed feature. Applicants show that the teachings of Bouchier are substantially different from the features recited in Claim 1. The Examiner cites Bouchier's service processor as being used to manage the partitions, and Bouchier's system firmware 901 running on processors of cell 0 and cell 1 in partition 0, and an ability of a user to run a debug client 902 on their workstation 901 to debug an operating system or firmware. It is important to note that Bouchier's debugger is used to debug system firmware 901 (see FIG. 1; Col. 12, lines 39-40), whereas Claim 1 is directed to debugging *partition management firmware*. The Examiner equates Bouchier's service processor 101 as reading on the claimed partition management firmware, but yet there is no ability to debug service processor 101 using Bouchier's debugger. While the service processor may be used to facilitate debug of the system firmware by passing commands between the user workstation and a cell (FIG. 8; Col. 12, lines 59-67), there is no teaching or suggestion of any ability to debug the service processor itself using the debugger. As the service processor is being equated with the claimed partition management firmware, and there is no ability per Bouchier to debug this service processor using the disclosed debugger, Applicants urge that the cited reference does not teach or suggest the claimed feature of "debug said partition management firmware". Applicants have amended Claim 1 to further clarify this distinction. Thus, Claim 1 is still further shown to not be obvious in view of the cited references.

Applicants initially traverse the rejection of Claims 2-8 for reasons given above regarding Claim 1, of which these Claims 2-8 depend upon.

<sup>1</sup> To establish *prima facie* obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974).

In the absence of a proper *prima facie* case of obviousness, an applicant who complies with the other statutory requirements is entitled to a patent. *See In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).

Further with respect to Claim 2, none of the cited references teach or suggest the claimed feature of "establishing function calls within said operating system debugger that when invoked by said operating system debugger cause said partition management firmware to execute partition management firmware operations". As can be seen, this claimed feature recites specific synergistic co-action between the operating system debugger and partition management firmware, whereby the operating system debugger has function calls that when invoked (by the operating system debugger) cause the partition management firmware to execute partition management firmware operations. Thus, the operating system debugger is advantageously able to indirectly access system resources controlled by partition management firmware by invoking function calls that cause the partition management firmware to execute partition management firmware operations. In rejecting Claim 2, the Examiner cites the Held reference as teaching a DSP debug API 441 that includes functions to control task execution. The Examiner then cites the Bouchier reference as teaching partition management firmware. Notably absent is any assertion of any co-action between the two, likely because each item is separately described in separate, non-related documents and hence these two items (Held's debugger API and Bouchier's partition manager) do not provide any synergistic co-action, including the specifically claimed feature of "establishing function calls within said operating system debugger that when invoked by said operating system debugger cause said partition management firmware to execute partition management firmware operations". Therefore, Claim 2 is further shown to not be obvious in view of the cited references.

Further with respect to Claim 3, none of the cited references teach or suggest the claimed feature of "reporting, from said partition management firmware to said operating system debugger, partition management firmware events in response to said operating system debugger invoking said cvcnt handler routine". For similar reasons to those given above regarding Claim 2, Held's debugger API and Bouchier's partition manager do not provide any synergistic co-action, including the specifically claimed feature of reporting, *from* said partition management firmware *to* said operating system debugger, partition management firmware events *in response to* said operating system debugger *invoking*

*said event handler routine.* Therefore, Claim 3 is further shown to not be obvious in view of the cited references.

Further with respect to Claim 4, none of the cited references teach or suggest the claimed feature of "directly reading data from a physical memory location utilizing said partition management firmware in response said operating system debugger invoking said read memory function call". Claim 4 further recites synergistic co-action between the partition management firmware and the operating system debugger, whereby data is read from physical memory utilizing the partition management firmware. In rejecting Claim 4, the Examiner cites Held column 9, lines 28-45 as teaching this claimed feature.

Applicants show that there, Held states:

"The target machine 540 issues commands, via the DSP debug module 442 and through the IA-SPOX host side VxD 405, to control a task being debugged using the debug task mailbox 620. Data, such as a task context or task memory contents, are exchanged via a stream. *The debug task mailbox 620, in one embodiment, is a portion of memory that enables multiple tasks to read from and write to the portion of memory.* Note that in one embodiment, the debug task mailbox 620 represents the same method of communications used between other tasks in the real-time environment 211. The task mailbox(es) 620 and streams used are facilities provided to all tasks launched through the DSP interface 450. That is, no special communications techniques were needed for tasks to communicate with the debug task 444. In one embodiment, the debug task mailbox 620 is implemented using standard streams and multiple standard mailboxes (e.g. one mailbox for receiving information and one mailbox for transmitting information)."

As can be seen, this passage merely describes controlling a task being debugged using the debug task mailbox which is a portion of memory that enables multiple tasks to read from and write to such memory. In contrast, Claim 4 recites specific synergistic co-action between a debugger and the partition management firmware to effectuate reading and writing to memory, and in particular Claim 4 recites that the reading from memory *utilizes said partition management firmware in response said operating system debugger invoking said read memory function call.* There is no teaching of this synergistic co-action in any of the cited references, and hence Claim 4 is further shown to not be obvious in view of the cited references.

Further with respect to Claim 5, none of the cited references teach or suggest the claimed feature of "directly writing data to a physical memory location utilizing said partition management firmware in response said operating system debugger invoking said write memory function call", the reasoning being similar to that given above regarding Claim 4.

Further with respect to Claim 7, none of the cited references teach or suggest the claimed feature of enabling/disabling debugging within said partition management firmware utilizing a service processor. This claimed feature advantageously provides, in addition to an operating system debugger and partition management firmware, a service processor wherein the service processor is used to enable/disable debugging within the partition management firmware. In rejecting Claim 7, the Examiner cites Held column 8, lines 31-37 and column 9, lines 4-10 and 52-58. Applicants urge that this passage merely describes that the debugger may suspend or resume task execution, and this task execution suspension/resumption is responsive to an exception event and based upon the context of the user's action. This is different from what is recited in Claim 7. A service processor is not used for suspension/resumption, as claimed, but rather an exception event. Therefore, as all claimed features are not taught or suggested by the cited reference, and in particular a service processor being used to enable/disable debugging within partition management firmware, Claim 7 is not obvious in view of the cited references.

Further with respect to Claim 8, none of the cited references teach or suggest the claimed feature of debugging partition management firmware that exists between the partition and the hardware. In rejecting Claim 8, the Examiner acknowledges that Held does not teach a partition management firmware that exists between partition and hardware, but states that Bouchier teaches partition management firmware exists between partition and hardware at Bouchier column 2, lines 27-28 and column 4, lines 60-65. Applicants show that the Bouchier passage cited at column 2 states that a service processor is used to manage the partitions and the hardware they run on. The Bouchier passage cited at column 4 discusses aspects of Bouchier FIG. 1, and specifically describes a partition having firmware, an operating system and application software running on the processors. Neither of these passages describes any ability to debug

partition management firmware that exists between the partition and the hardware. For example, there is no ability to debug the service processor described at column 2, and the firmware described by Bouchier column 4 is not partition management firmware. Therefore, Claim 8 is further shown to not be obvious in view of the cited references, as there are missing claimed features not taught or suggested by the cited references.

Further with respect to Claim 8, such claim has been amended to emphasize the debugger's ability to directly access the physical memory of the data processing system using the debugger extensions in conjunction with said partition management firmware. None of the cited references teach or suggest such capability, which advantageously provides an ability for the debugger to access physical memory and thereby enable debugging of the partition management firmware (Specification page 14, line 25 – page 15, line 12).

With respect to Claims 9-16, Applicants traverse for similar reasons to those given above regarding Claims 1-8.

With respect to Claims 17-24, Applicants traverse for similar reasons to those given above regarding Claims 1-8.

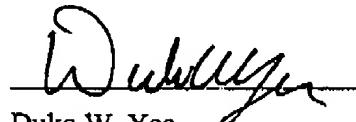
Therefore, the rejection of Claims 1-24 under 35 U.S.C. § 103(a) has been overcome.

**II. Conclusion**

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 7/29/04

Respectfully submitted,



Duke W. Yee  
Reg. No. 34,285  
Wayne P. Bailey  
Reg. No. 34,289  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 367-2001  
Attorneys for Applicants